

THE

QL

MAGAZINE

Volume 31 Issue 3: JUN - JUL

QUANTA 2014

Hardware for the QL  
coming together?



**QUANTA**  
QL USERS & TINKERS ASSOCIATION

# QUANTA

## INFORMATION ON THE ASSOCIATION

Membership of QUANTA, the independent QL user group, is by annual subscription. The Membership Secretary can supply full details. Copies of the association's constitution & annual accounts are available from the Secretary.

### QUANTA Committee - 2014/2015

Chairman: Sarah D. Gilpin 181, Urmston Lane, Stretford, Manchester. M32 9EH

Email: [chairman@quanta.org.uk](mailto:chairman@quanta.org.uk)

Tel: 0161 865 2872

Secretary: Alison Southern 40, Distaff Road, Poynton, Cheshire. SK12 1HN

Email: [secretary@quanta.org.uk](mailto:secretary@quanta.org.uk)

Tel: 01625 850067

Treasurer & Membership Secretary: John Gilpin 181, Urmston Lane, Stretford, Manchester. M32 9EH

Email: [treasurer@quanta.org.uk](mailto:treasurer@quanta.org.uk)

Email: [membership@quanta.org.uk](mailto:membership@quanta.org.uk)

Tel: 0161 865 2872

Magazine Editor: Lee Privett. 25 Mucking Hall Road, Barling Magna, Southend on Sea, Essex. SS3 0NH.

Email: [editor@quanta.org.uk](mailto:editor@quanta.org.uk)

Mobile: 07985 513 234

Librarian: David Buckley 136, Denton Lane, Chadderton, Oldham. OL9 8PS

Email: [librarian@quanta.org.uk](mailto:librarian@quanta.org.uk)

Tel: 0161 622 1600

Helpline & News Editor: Dilwyn Jones 22 Erw Las, Coetmor New Road, Bethesda, Gwynedd. LL57 3NN

Email: [helpline@quanta.org.uk](mailto:helpline@quanta.org.uk)

Email: [news@quanta.org.uk](mailto:news@quanta.org.uk)

Tel: 01248 601599

Web Administrator & Web Master: Keith Dunbar 44, Dalton Avenue, Stretford, Manchester. M32 9TP

Email: [webadmin@quanta.org.uk](mailto:webadmin@quanta.org.uk)

Mobile: 07789 483 800

I.C.M.: Dave Park, 1406 Dana Court, Round Rock, TX USA 78664

Email: [dave@quanta.org.uk](mailto:dave@quanta.org.uk)

Tel: (512) 529 7863

Back Issues Requests: Basil Lee 31 Fairdale Gardens, London. SW15 6JW

Email: [backissues@quanta.org.uk](mailto:backissues@quanta.org.uk)

Tel: 0208 789 1976

*QUANTA reserves the right to publish, reject or edit any material submitted. The opinions expressed are solely those of the contributors. Member's private small ads - free (up to 50 words). Non-members private small ads (and members after 50 words) 50p per 10 words or part thereof. Terms strictly paid up-front with order. Receipt if requested. All copyrights and trademarks are hereby acknowledged.*

# CONTENTS

EDITORIAL .....	LEE PRIVETT .....	4
QUANTA NEWS.....	DILWYN JONES .....	5
QUO VADIS DESIGN .....	ADVERTISEMENT .....	8
RWAP SOFTWARE .....	ADVERTISEMENT .....	16
CHAIRMAN'S NOTES.....	SARAH GILPIN.....	20
FLOATING POINT ARITHMETIC	GEORGE GWILT .....	21
JOCHEN MERZ SOFTWARE.....	ADVERTISEMENT .....	22
SANDY ELECTRONICS PT2 .....	DAVE PARK.....	33
UPDATED REPORT .....	JOHN GILPIN.....	37
QUANTA HELPLINE.....	DILWYN JONES .....	38
RWAP MEMBRANES .....	ADVERTISEMENT .....	40
COPY DATE .....	(FOR NEXT ISSUE) .....	51
QL FORUM .....	ADVERTISEMENT .....	52

Visit our website at : [www.QUANTA.org.uk](http://www.QUANTA.org.uk)

'Like' us on Facebook at : [www.Facebook.com/](http://www.Facebook.com/)

[QUANTA.org](http://QUANTA.org)

Currently watching this video? [QL is 30 - The Collection \(Sinclair QL 30th launch-day anniversary\)](#)

**N**othing seems to have changed in my household, my QL stuff is still packed and unseen, I have the sinking feeling of at least a year maybe two before I get back to what I would call normal QL'ing. Suffice to say I am recovering from a ladder fall and other than the occasional peruse on line the rib repair is taking its time and painful!

The last time I cracked my ribs was in Ibiza 1982 playing football with the staff of the hotel where I literally ran in to a wall, I kid you not. New skills I have learned recently are; how to cough whilst hugging a wall/ cushion/pillow; I know 16 other sleeping positions and I can convert a sneeze in to an expletive half way through. 1982 was two years before the QL came on to the scene and now 30 odd years later I have done it again, will I never learn?

One of the celebrations of '30 years of the QL' will occur in Edinburgh early October, partly sponsored by QUANTA and many other QL enthusiasts, we hope that lots of other QL supporters will make the effort of what is likely to be the last main celebration of this fine ahead of its time computer.

This is unlike the QL scene where things seem to be moving forward a bit in a positive way, both in hardware and software. There has been numerous comments on whether development in hardware is relevant or needed and emulation is the only way to go. Well what are your thoughts? Do you want QUANTA to continue in the way it currently does? Do you want to see changes? Do you want QUANTA to continue? If you have an opinion strong or indifferent let us know, contact details as always are in the magazine.

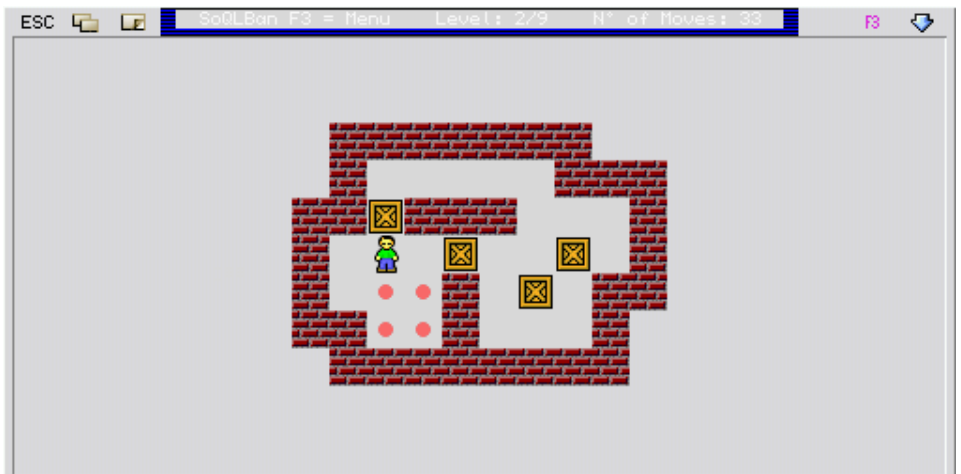
*The Editor*

If you have QL-related news items that you'd like us to include on this page, please get in touch with News Editor - Dilwyn Jones at [news@quanta.org.uk](mailto:news@quanta.org.uk)

**SoQLoban**

Wolfgang Lernerz has released a new QL game called SoQLoban. This is a version of a game called Sokoban, where you take the part of a warehouse-keeper (the word Sokoban apparently means Warehouse Keeper in Japanese) who has to push crates to their intended storage locations in a warehouse, which sometimes looks more like a maze.

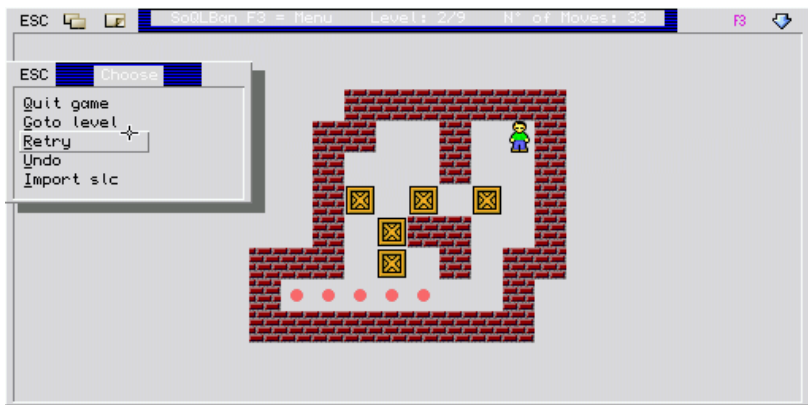
Storage locations are marked on the floor with a coloured dot. The Warehouse Keeper has to push the crates horizontally or vertically (get behind a crate and press an arrow key) until the crate is over one of the dots. Sounds easy, at first it is, then it gets harder.



The game ends when all the crates have been moved to these storage locations, then proceeds to the next level. There are nine puzzles built in, and Wolfgang has included a few extra levels files which can be imported into the game (files with names ending in \_slc)

To start the game you right click (or press ENTER) somewhere in the game window. The cursor keys then become active to move the little Warehouse Keeper around so you can start pushing the crates around.

You can press F3 to get a small menu of options to let you quit, go to a level, retry, undo one push (handy when you realise you just blocked yourself in a dead end or stack more than one crate in a manner which makes it impossible to move them), or import as described above.



The game makes good use of colour and the colour ways can be configured with MenuConfig. The game is free and available to download (sources also available) from Wolfgang's website at:

<http://www.wlenerz.com/qlstuff/>

COMPLAINT: This news column was nearly late because your News Editor got addicted to playing this game.

## **SMSQ/E v3.21**

SMSQ/E 3.21 is out.

This contains a workaround for a (rare) bug in the WIN driver. You might want to upgrade. The list of changes on Wolfgang's website includes the following:

Added file open check when making directories.  
SMSQ/E DV3 floppy driver for SMSQmulator.

Download the SMSQ/E binaries from:

[www.wlenerz.com/smsqe](http://www.wlenerz.com/smsqe)

## **SMSQmulator v2.02**

Wolfgang Lenerz has also released SMSQmulator 2.02.

Mostly bug fixes for the floppy driver – the list of changes says it can now handle read-only floppy image files, and includes a 'kludge' for C programs which try to write to address \$10 and if so make wrong assumptions about the machine. See the full list of changes on the same page as the downloads:

<http://www.wlenerz.com/SMSQmulator/>

## **Backup**

As the name implies, this is a backup program. Another release from Wolfgang Lenerz. Can backup a whole drive or just a selected directory if required. The program needs the Menu Extension. In

QUO VADIS  
DESIGN

Independent Information  
Technology Services

www.ql-qvd.com

QL/QDOS/SMSQ/E Software

QUO VADIS DESIGN Independent Information Technology Services

QL/QDOS/SMSQ/E Software

Home Products Support Company Contact

Welcome

Quo Vadis Design sells software for the Socratic Quantum Leap computer (QL) and variants including a new OS called SMSQ/E.

The QL is a computer in its 25th year Anniversary.

News

- QVD QL News Blog - keep up to date  
[News Blog](#)  
24/04/2009
- Quo Vadis Design Website Launched  
01/02/2009

FEATURED PRODUCT



BUY NOW!

Copyright © 2009 Quo Vadis Design. All Rights Reserved. Home | Products | About | Contact | Order

Bruce@ql-qvd.com

Quo Vadis Design  
38 Derham Gardens  
Upminster  
RM14 3HA  
UK

Tel: +44 (0)20 71930539  
Fax: +44 (0)870 0568755

**QPC2 IS NOW FREE AND  
AVAILABLE FROM MARCEL  
KILGUS WEBSITE**

<http://www.kilgus.net/qpc/index.html>

**Congratulations to Marcel  
on becoming a Father**

**Check the QL News Blog on  
our website for updates.  
[www.ql-qvd.com/blog](http://www.ql-qvd.com/blog)**



what is probably a first for QL software, this program also shows it's possible to have tab pages and clickable column headers. Download the program and sources from Wolfgang's page at :

<http://www.wlenerz.com/qlstuff/>

Name ↓	Date	Size
basic_backup_bas	2014.04.26 17:15	115590
basic_backup_blnk	2014.04.19 17:13	1554
basic_backup_changesize_bas	2014.03.07 11:34	1836
basic_backup_checkoother_bas	2013.04.06 09:08	2150
basic_backup_comparedir2_bas	2014.03.13 10:27	3700
basic_backup_comparedir_bas	2014.04.25 07:08	4267
basic_backup_compare_bas	2014.04.25 06:22	3715
basic_backup_data	2014.03.20 11:08	534
basic_backup_data_bin	2014.03.20 11:17	1120
basic_backup_doc	2014.03.21 10:35	4118
basic_backup_do_sort_bas	2014.03.13 10:01	2679
basic_backup_drawsmalappsub_bas	2014.04.25 06:22	1775
basic_backup_drawsortappsub_bas	2014.03.15 09:43	817
basic_backup_errors2	2013.03.20 17:38	68
basic_backup_initold_bas	2013.03.10 10:49	4385
basic_backup_main_bas	2014.04.26 15:39	4181
basic_backup_make_appsub_bas	2014.03.10 10:30	1934
basic_backup_mkdat_bas	2014.03.05 14:03	166
basic_backup_months_bas	2014.03.05 22:16	435
basic_backup_obj	2014.04.26 17:15	82176

### S\*BASIC Keywords List

Bob Spelten has overhauled, updated and completed the S\*BASIC keywords list made initially by François van Emelen. The new list is on the SMSQE site ( [www.wlenerz.com/smsqe](http://www.wlenerz.com/smsqe) ), with additional information and data.

### QPTR Manual

The latest version of the QPTR (Pointer Environment) manual is out (6<sup>th</sup> edition at the time of writing). It hopefully is up-to-date with the latest developments, and available as a PDF or LibreOffice ODT file. You can find it at [www.wlenerz.com/smsqe](http://www.wlenerz.com/smsqe) and more directly at <http://www.wlenerz.com/smsqe/add1.html>.

## **QL Meeting in Modena, Italy**

Between those who attended in person and electronically via Skype, nearly 20 QL users came together for a meeting in Modena. This was the first Italian meeting of recent years and given that it was organised at fairly short notice, the turnout was quite respectable.

The highlight of the afternoon was Dave Park of Sandy Electronics giving a talk about his forthcoming QL products. English/Italian.

Translation was provided by Michele Perini by Skype from Toronto. People from several countries listened in by audio, and could see a video feed of the meeting itself. Davide Santachiare, former QL trader Ergon Development, admitted he had been concerned (as he was not a Skype expert) about how well he expected this to work, but apart from a few very minor glitches it worked remarkably well and showed just what can be achieved.

Quanta has used these types of media for committee meetings and AGM, so the QL scene is gathering experience and this may well become a fixture of future QL meetings, such as QL Is 30 in Edinburgh this autumn.

Davide hopes to provide a summary report of the Italian meeting on his website at <http://www.sinclairql.it/>

## **Qubide Tools Repository**

Graeme Gregory has set up a github repository for the Qubide Tools program he has been working in – find it at:

<https://github.com/slimlogic/qubide-tools>

## Homebrew Hardware

It would appear that Dave Park's work on producing new QL hardware such his proposed Super-RAM and Ultimide for example has produced a bit of a rush to make DIY QL hardware add-ons. Phill Harvey-Smith, who has also been working on furthering the MESS QL emulator, has been working on a replacement QIMI card with add-ons. Here is a picture of his second prototype:



On May 1st, Phill wrote: *“This has TQFP chips to allow the board to shrink to just using the ZX8302 and right hand ROM socket. This allows the QL SD to plug into the left.*

*The board supports either a single 28 pin 27512 EPROM with the ROM version of your choice, or a 32 pin flash ROM / EPROM with up to 8 banks of rom, which are mapped into the lower 64K. ROM*

*banks are selected by poking the bank number into 114590 thus :*

*bits 0..2 : bank number 0..7*

*bit 4 if 0 then the rom ins not mapped into \$c000-\$ffff*

*(default), if 1 then it is. This allows the cartridge rom to be on the built in rom or on a physical cartridge.*

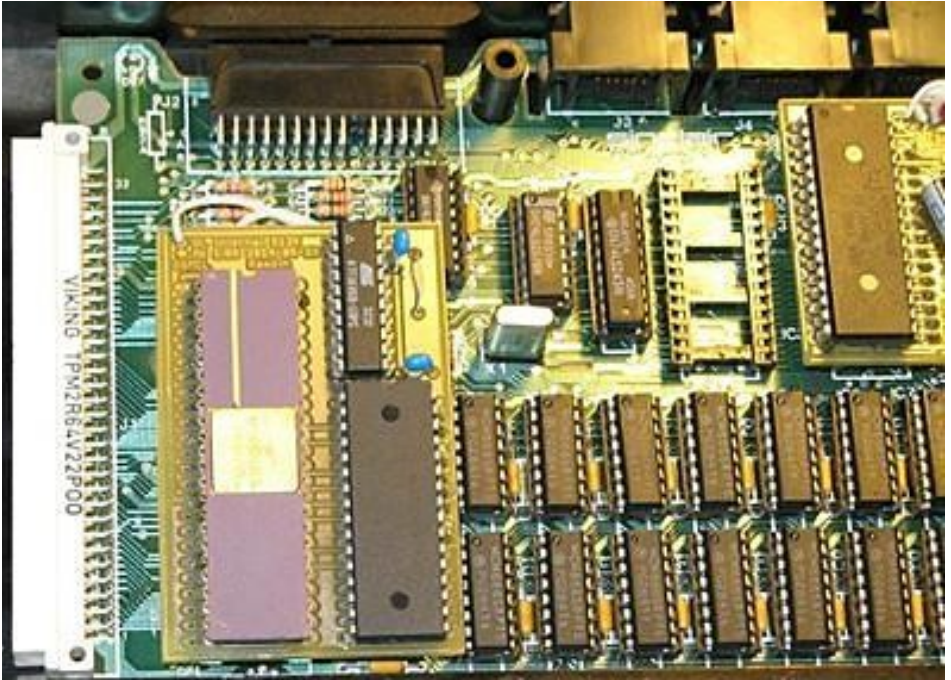
*The new ROM bank is made available next time that the reset is pressed, this allows for several versions of the rom to be present in the machine and software selectable e.g. Minerva and JM (for example)...even TYCHE works as long as you enable the cart rom for it's bank...*

*Next step is to be to get a small number of professional boards made to further test the design.”*

Phill has also described how this board can use a PS/2 mouse via an on-board converter, rather than a rarer and more expensive Atari mouse used by the original QIMI.

Not yet clear if this would ever become a commercial project for the QL, but it's highly interesting nonetheless, especially as it brings a QIMI up to date and offers other benefits, such as multiple ROM images, PS/2 mouse and ability to co-exist with a QL-SD inside the QL.

Meanwhile, a user on QL Forum who goes by the name 'tcat' has produced his first hardware add-on, a 512K add-on memory board, and another user (who says he has a total of 7 QLs) went on to produce an internal version of the board.



In common with other internal expansions, this one has a little white wire you can see at the top of the board which has to go to the DSMCL signal, as that's not available on the CPU socket into which this board plugs.

What's interesting is that these users have chosen to make the designs freely available so that those with the knowledge and experience of the assembly of projects like this can build their own. Plus, they have been using the QL Forum as a discussion point to share views and experience and to help each other with these projects.

Follow the discussion and news of progress on the Hardware section of QL Forum, at <http://qlforum.co.uk/viewforum.php?f=2>

## **QL Is 30, Edinburgh**

The QL Is 30 event has now been booked for 11<sup>th</sup> October 2014 in Edinburgh, Scotland. The venue will be Gyle Hall of the St. Thomas Episcopal Church, Glasgow Road, EH12 8LJ. The event is being organised by members of the Scottish QL Users Group (SQLUG) assisted by Geoff Wicks.

The venue is on the main road from Edinburgh city centre to Edinburgh Airport and Glasgow. The bus service to and from the airport has a bus stop outside the church.

Geoff Wicks is planning to visit Edinburgh in early June to do some research and photography, and then hopes to set up a dedicated website for the event.

This should be a truly international meeting, with expressions of interest already from QL users in several countries. Geoff has made one travel suggestion, that *“any one wishing to come by car [from the continent] may like to consider the overnight DFDS ferry from IJmuiden in the Netherlands to Newcastle. This would save a long drive in the UK. Any one thinking of travelling by Eurostar should note that the London terminal is next door to Kings Cross station with trains to Edinburgh.”*

## **QL at “Geek Is Good”**

Stephen Usher writes:

*“In case any of you are interested one of my QLs will be on display as an interactive display in the upcoming ‘Geek is Good’ exhibition at the University of Oxford, Museum of the History Of Science starting on the 15th May and going on until September. It'll be sitting there connected to a Microvitec monitor with the Sandy Qboard and a floppy drive, along with the manual.*

*So, if you want to pop along and amaze people with your SuperBASIC knowledge or squirrel in a floppy disk to boot from you know where to go. On one evening there will be a special event, I think on Saturday, during which the QL will be running Jochen Merz's 'Arcanoid II' game next to a BBC Micro running something else."*

The exhibition traces "geek" from ancient times showing that they were often the innovators who lead to today's scientific and technical world.

More information and (eventually) a gallery at:

<http://www.mhs.ox.ac.uk/>

### **Ultrasoft Programs Now Available**

A number of formerly commercial Ultrasoft utility programs are now available as free downloads from the Sinclair QL Homepage, following Rich Mellor's communication with Martin Berndt.

**Disk Monitor 2.4.** Doctor your disks. Rescue corrupted disks. Alter programs. Edit directories. User friendly. HEX and ASCII on screen entries for file or sector editing. Former commercial program from Ultrasoft. (10K download).

<http://www.dilwyn.me.uk/files/index.html>

### **Q-Kick 4.1.**

The following description is from their advert. *"Q-Kick is the program for the user who needs to work. A fully multitasking program running completely in interrupt 7 as a call-up background routine. Complete desk top facilities including a notepad and*

# RWAP Software

We have been working hard to safeguard the future of the Sinclair QL through creating an online QL Wiki backed by a repository of QL titles.

The QL Wiki enables people to see what software and hardware was produced for the Sinclair QL. The repository enables people to purchase working copies when their microdrive cartridges fail, or they move onto disks or emulators. Where we can, we obtain permission from the copyright holders to re-release items on the market.

This has enabled us to bring some of the titles back to the market as well as re-ignite interest in the Sinclair QL and its emulators.

Old favourites brought back to life include:

E.V.A., Vroom  
Black Knight, Double Block  
Jungle Eddi, The Lost Pharaoh  
Pangi, the Lonely Joker  
Microdeal's Flight Simulator, QL Hopper, The King,  
Night Nurse and Cuthbert in Space  
Mortville Manor, Darkside of the Moon (£10)

We still retain our existing catalogue, including:

Open Golf, Return to Eden,  
Stone Raider II, The Prawn,  
Hoverzone, Lost Kingdom of Zkul,  
Deathstrike, Horrorday,  
Flightdeck and QWord

All games are available on disk or for use with Q-emuLator on the PC **from ONLY £5 each**

Microdrive versions also available - from £10 each.  
*Memory / system limits may apply - please check before ordering.*

Visit the QL Wiki for more details on software, books and hardware for the Sinclair QL:

[www.rwapadventures.com/ql\\_wiki](http://www.rwapadventures.com/ql_wiki)

**RWAP Software**  
3 Dale View Court, Fulford, Stoke-on-Trent  
Staffordshire ST11 9BA  
Tel: 01782 398143  
Email: [rich@rwapservices.co.uk](mailto:rich@rwapservices.co.uk)

Website:  
[www.rwapsoftware.co.uk](http://www.rwapsoftware.co.uk) or [www.sellmyretro.com](http://www.sellmyretro.com)



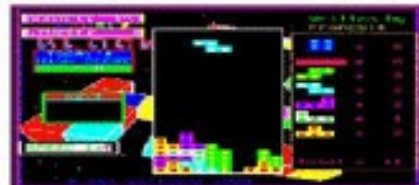
Vroom by Pyramide



E.V.A by Westwey



Mortville Manor by Pyramide



Double Block by Francois Lancaut



Night Nurse by Microdeal



*appointment diary. Save screens while playing - ideal for hi-score proof etc. Save memory, load memory, alter job priorities and much more for the programmer while working within the current program.*

*For the businessman there is not only the note-pad and appointment diary and calendar but an easy to use database for addresses etc., with a software interface to the Psion Organiser II and with an easy transfer routine to put those addresses or the data you need into the current program, i.e. while working in Quill, find the addresses you need and transfer it straight into Quill from Q-Kick or quickly look up an address or telephone number while working with any other program.*

*It also comes with an automatic screen switch on/off routine to prolong the life of your monitor and a real multitasking spooler - spool from within Quill". This includes a plain text manual (QKick\_txt), (26K download).*

<http://www.dilwyn.me.uk/utills/index.html>

**UltraUtils:** an assortment of small programs from Ultrasoft. Includes Digiclock, FontEditor, Pickall (pick programs in reverse order), PrinterPanel, SystemInfo and SystemPanel. Note that some of these programs won't work on pointer environment version 2.00 or later, as the PE version test only tests the last two digits of the version number (e.g. digiclock tests for version x.14).

The assembler sources are available from the same page if anyone wants to have a go at correcting and reassembling this! In the meantime, if you'd like to try out these little utilities, you can download earlier (pre-v2.00) versions of the pointer environment

files ptr\_gen, wman and hot\_rext from my Pointer Environment page (program: 16K, sources: 115K download).

<http://www.dilwyn.me.uk/utils/index.html>

Additionally, I have added an English version of Ultrasoft's MF ROM (enhanced MG ROM) to the QL ROMs page on my website at <http://www.dilwyn.me.uk/qlrom/index.html> and TurboXchange, a largely unofficial version of Xchange, which is faster and slicker than the original, is also available from the Psions page on my website at: <http://www.dilwyn.me.uk/psions/index.html>

### **QL Cadette**

Another formerly commercial program, QL Cadette, is now available to download free from my website at :

<http://www.dilwyn.me.uk/graphics/index.html>.

QL Cadette is a compact two dimensional drafting tool for Sinclair QL, including unexpanded 128K systems. An accompanying utility, Plot, allows drawings created with the help of Cadette to be drawn with a plotter, or, with Epson compatible dot matrix printers fully corrected screen dumps may be generated - circles are circles, not ellipses. A plain text manual is included in the archive, and a PDF manual is available below. Mark Mansell at Bestmalt has given permission for this former QL software to be released as freeware.

### **SMSQ/E Manuals**

V1.02 of the updated SMSQ/E manual is now available from the eBooks page on my website. The most recent versions (available in a variety of formats) and now include Word-type Table of Contents and Index at the start and end, to make updating easier.

Download from <http://www.dilwyn.me.uk/docs/ebooks/index.html>

## **French QL Manuals**

Olivier Basely has now completed his work on scanning and reproducing the French edition of the QL manual, which now includes the Introduction, Beginners Guide, Keywords, Concepts, Quill, Archive, Easel and Abacus manuals, all available to download from <http://www.dilwyn.me.uk/docs/ebooks/index.html>

## **PD Software Updates**

A very large number of older PD and Freeware QL programs have been added to my site in recent weeks. The list of programs is far too long to include here – see <http://www.dilwyn.me.uk/history.html> Much of this activity has been down to just one person – Martin Head. He volunteered to help me,

I sent him a CD with a huge amount of old QL software which needed going through and bless him, ever since he's been sending me weekly zip files ready to paste onto my website. I'm extremely grateful to him for his hard work which has allowed me to make so much old software available once more.

Rich Mellor has also put in one heck of a lot of work in locating, contacting and negotiating with authors and publishers of older software as part of his drive to preserve old QL software. Thanks to Rich's work, I am now able to offer important software such as Prospero's Fortran compiler, business software from SD Microsystems and most of the Ultrasoft range of QL software. Thank you too, Rich.

## **QL Wiki**

Rich Mellor would like to appeal for someone to help him with

maintaining the QL Wiki, something he does not have the time to manage at the moment. Rich writes:

*“This year marks the 30th anniversary of the Sinclair QL - we should be making the most of any publicity surrounding the anniversary, and promote the QL, its hardware and software as much as possible, so as to get some impetus behind the various projects going and maybe tease out some of the copyright holders, so that more software can be re-released.*

*The Wiki is crucial to this effort - there is masses of information which could be added to it - by anyone with half an hour to spend looking at a QL World / QL User magazine and entering some of the information from an advert (say the Digital Precision adverts which had a mass of information / hype about their programs).”*

So if anyone would like to volunteer their time to do this, please contact Rich at RWAP Services. The QL Wiki is like most other Wikis – you can get your own login to edit the text online. You will be doing the QL community a great favour in keeping this important source of information up to date.

If you have read the list of updates to my website (see ‘PD Software Updates’ above) you will have seen just how much activity and effort has gone into tracking down, cataloguing and contacting authors and publishers behind the scenes.

## **CHAIRMAN’S NOTES**

**SARAH GILPIN**



A new year has begun for QUANTA and we look forward to the celebrations planned for QL is 30.

The Annual General Meeting was held on Sunday 26<sup>th</sup> April 2014, with eleven people attending at 181, Urmston Lane and four attending by Skype. We also had six proxy votes which gave us twenty one voters, the same number that were recorded in 2013. We had apologies from Rich Mellor, who was unable to attend on line. As there were not enough nominees to fill the number of positions on the Committee, all nominations were accepted unopposed.

I am delighted to welcome our first American committee member, Dave Park. I wish to thank all the Committee members for their commitment to QUANTA and the QL community.

It is with regret that we have accepted John Gilpin's resignation as Treasurer and Membership Secretary. We are urgently looking for someone to take on these roles.

John and I will be attending the event in Edinburgh on October 11<sup>th</sup> 2014, and so will a number of the NEMQLUG members. The subgroup is planning to provide something for the event. We hope that all the Sub-groups will be able to send a representative.

## **FLOATING POINT ARITHMETIC                      GEORGE GWILT**

**P**erforming floating point arithmetic on the QL using Assembler Language is made easy by two vectors, RI\_EXEC and RI\_EXECB. These work by being presented by bytes whose values indicate the operations to be performed.

Arithmetical operations are performed on numbers held on the arithmetic stack. There is also a data stack used for storing numbers.

**JOCHEN MERZ SOFTWARE**

**Kaiser-Wilhelm-Str. 302  
47169 Duisburg, Germany**

**Fax +49 203 502012  
EMail: SMSQ@J-M-S.com**

**SMSQ.J-M-S.COM**

**Thanks to Marcel, QPC2 is now freely available.  
You can download it from Marcells homepage**

**[www.Kilgus.net](http://www.Kilgus.net)**

**If you wish to print from QPC2, then you need**

**QPCPrint**

**... which is available from J-M-S.**

**Only 39.90 if you choose EMail delivery.**

**For additional 4 EUR, delivery will be on CD.**

**QPCPrint will allow you to print to (more or less) every printer which is installed under Windows (dot matrix, ink, laser, PDF "printer", FAX "printer" etc.)**

**You can place your order via letter or use the SSL order form on SMSQ.J-M-S.COM - click on "Online orders".**

**If you order by Mail or through the SSL contact form: We now accept VISA, MasterCard, Diners Club, JCB, Discover, UnionPay, BCard, DinaCard and American Express!**

**The order form has not been updated for the new card types at the time I write this ad, but it will be updated soon.**

**I am still working on changing the forms and implementing SEPA on all the order forms.**

As an example, the byte `qa.mul`, whose value is `$0E`, causes the top two items on the arithmetic stack to be multiplied together. Bytes with values between `$33` and `$FF` transfer numbers between the arithmetic stack and the data stack.

The names, such as `qa.mul`, of the operations with codes up to `$32` are given in section 18.10 of the QDOS Reference Manual.

### ***Where are the Stacks?***

The locations of the bases of the arithmetic and data stacks are indicated to the vectors by registers `A1` and `A4` respectively. In each case the values must be relative to `A6`. Thus the arithmetic stack is at `(A1,A6.L)` and the data stack at `(A4,A6.L)`. The stacks can, of course, be set up anywhere. One way is to reserve space inside the program itself. Another is to grab space from the Heap by using the Trap #1 call, `MT_ALCHP`. In these cases the values in both `A1` and `A4` can be made relative to `A6` either by zeroing `A6` (a dramatic move?) or by subtracting `A6` from `A1` and `A4`.

I prefer another approach. I use the program's own data space. By setting the instruction

```
LEA      (A6 ,A4 . L) ,A6
```

near the start of the program I set `A6` pointing to the data space.

The commands `RSSET` and `RS.x` (in `GWASS`) can be used to set the value of labels of items in data space relative to `A6`. Thus:

```
; SET_UP  
; Defines the space for five objects and gives
```

```
; values to the labels
; Thus ITEM2 = 6 for example
```

```
        RSSET    0    ; sets the first label's
                    ; value to 0
ITEM1    RS.W     3    ; allocates a 6-byte space
ITEM2    RS.W     3    ;
ITEM3    RS.W     3    ;
vars     RS.B     40   ; allocates 40 bytes for the
                    ; arithmetic stack
a_st     RS.B     1    ; sets the address of the
                    ; arithmetic stack
```

In this simple example the value of vars, which is the base of the data space, will be 18, and of a\_st, the base of the arithmetic stack, 58. These are the values to set in A1 and A4 for the vectors RI\_EXEC and RI\_EXECB.

### ***Transference Between Stacks***

ITEM3 on the data stack starts 6 bytes before the base of the stack. Thus, to transfer ITEM3 to the arithmetic stack we must arrange to access the address  $-6(A4, A6.L)$ . The byte code to do this is \$FA. The byte codes for ITEM2 and ITEM1 are \$F4 and \$EE. Adding 1 to these byte values gives the codes for transferring from the arithmetic stack to the data stack.

As an example, the codes for adding ITEM1 to ITEM2 and putting the result in ITEM3 are:

```
; OP_LIST
```



```

$EE      ; ITEM1 to arithmetic stack ($EE = -18)
$F4      ; ITEM2 to arithmetic stack ($F4 = -12)
qa.add   ; Add
$FB      ; Put the result to ITEM3 ($FB = -6 + 1)

```

In this set of four bytes, only qa.add indicates what is happening.

I reckoned that it would be nicer if the other three bytes could be replaced by rather more informative labels. Thus I would like \$EE to become g\_ITEM1, \$F4 to become g\_ITEM2 and \$FB to become p\_ITEM3. In other words, I would use g\_.. to get an item from the arithmetic stack and p\_.. to push it to the data stack.

Being an essentially lazy person, I devised a macro to do this. I also arranged that it would set up the data and arithmetic stacks as well.

Coding for the above example, using this macro, is:

```

; For SET_UP
; In addition this defines the labels p_...
; and g_... used in the OP_LIST

        maths_a   ; Sets the start of the process
        maths     ITEM1
        maths     ITEM2
        maths     ITEM3
        maths_e   40

; The OP_LIST becomes:

; OP_LIST

```

```
g_ITEM1
g_ITEM2
qa.add
p_ITEM3
```

The "maths" macros to do this is are given below.

```
; math_set_asm
; macros for RI_EXECB

; Start with "maths_a"
; followed by "maths item" for each item
; and end with "maths_e stacksize"
; This sets the rs.w values for each item
; and sets
; g_item for variables to stack (get)
; p_item for stack to variables (push)
; vars is set to the variables base
; (relative to A6)
; a_st is set to the base of the arithmetic
; stack (relative to A6)
```

```
maths_a    macro
__a        set        0
           endm

maths      macro      name
\1         rs.w      3
g_\1      equ        __a-__b+$100
p_\1      equ        g_\1+1
```

```

__a      set      __a+6
        endm

maths_e  macro    stacksize
__b      set      __a      ; __b is set to the
                               ;end of the last item

vars     rs.b     \1
a_st     rs.b     1
        endm

```

The value given in the macro "maths" for g\_\1 might not be immediately obvious. If the size of the data stack is S and the position of an item in the stack is P, then the address of the item will be S - P bytes earlier than the stack's base, which is in A4 relative to A6. The address of the item is thus:

$$-(S - P) (A4, A6.L)$$

The byte code for that address is thus \$100 - (S - P), or P - S +\$100. In the macro the value of P is \_\_a and of S, \_\_b. This explains the line:

```
g_\1     equ      __a-__b+$100
```

You might note that \_\_b, the size of the data stack, is set in the macro "maths\_e", which comes after the set of "maths".

### ***A Restriction Circumvented***

The codes available for getting and pushing variables allow for no more than 34 floating point numbers on the data stack, which is thus limited to 34\*6, or 204, bytes. For many purposes this is quite

enough. However there will be cases where the number of items exceeds 34. For example the addition of two  $n \times n$  matrices to produce a third would require  $3n^2$  numbers.

The data stack is only big enough to do this for matrices up to  $3 \times 3$ . For matrices larger than that the numbers involved will have to occupy more than one data stack. The outline of a solution to this problem is to identify each number by both its position in its own data stack, as in the above example, and also by a code indicating that particular 204-byte data stack, of which there may be several.

If all the items recorded are set one after the other in the program's data space the first 34 floating point items will fill the first data stack and the next set, the second data stack and so on.

Thus, in addition to  $p_{...}$  and  $g_{...}$  I would have  $w_{...}$  which would take values 0, 1, 2 etc. for the 1st, 2nd, 3rd etc. data stacks.

How could such a system be implemented? Since using a different data space requires altering the pointer in A4 it seems that RI\_EXECB could not be used.

My proposal is that RI\_EXEC be used instead and that a subroutine go through the OP\_LIST one byte at a time. This would allow A4 to be reset between any two operation bytes. In this case the OP\_LIST shown above would become:

```
; OP_LIST (multiple data spaces)
```

```
g_ITEM1
```

```
w_ITEM1
```

```
p_ITEM2
```

```
w_ITEM2
qa.add
p_ITEM3
w_ITEM3
```

To produce the new items w\_... we will need a new macro. I also assume that in this scenario there will be a set of at least one data stack of length 204 bytes, so that each can contain exactly 34 floating point numbers.

The new macro is:

```
; math_set2_asm
; macros for RI_EXEC

; Start with maths_a
; followed by maths item for each item
; and end with maths_e data stacksize

; This sets the rs.w values for each item
; and sets -
; g_item for variables to stack (get)
; p_item for stack to variables (push)
; w_item for the particular data set
; vars is set to the base of the 1st data stack
; a_st is set to the base of the maths stack

; Note - only 34 items can be got or pushed.
; So the nth item of data is accessed
```

```

; by p_ or g_ with w_
; If  $n = 34k + r$ , p_ and g_ have the
; value for r and w_ has the value k.

```

```

maths_a  macro
__a      set          0
__size   set          0          To count 34 items
__nmb    set          0          To give the set of
                                   items (0, 1 ..)
                                endm

maths    macro        name
\1       rs.w         3

g_\1    equ          __a+$34      $34 = $100 - 34*6
p_\1    equ          g_\1+1
__size   set          __size+1

                                if __size=34

__size   set          0
__a      set          0
__nmb    set          __nmb+1

                                endif

w_\1    equ          __nmb
__a      set          __a+6
                                endm

```

```

maths_e    macro      arithmetic stacksize
              if __a <> 0
spr       rs.b       204-__a      Complete the
                                      last data space
              endif
vars     rs.b       \1         The distance to
                                      arithmetic base
a_st     rs.b       1
              endm

```

### *Using RI\_EXEC*

Normally one would use the following code to activate an OP\_LIST.

```

lea      a_st,a1           -> arithmetic stack
lea      vars,a4          -> data stack
lea      op_list,a3       -> operations
movea.w  ri_execb,a2
jsr     (a2)

```

To use RI\_EXEC in place of RI\_EXECB so that the operations w\_..., which alter the value of A4, can be obeyed, the last two instructions can be replaced by the single one:

```
bsr      do_execb
```

Code for do\_execb is given below.

```
; do_execb uses RI_EXEC to process a
```

```

; string of operations
; On entry A1 -> arithmetic stack relative to A6
;           A3 -> the string of operations
;           A4 -> base of 1st (or only)
;                   204-byte data stack
; On exit  D0 contains error OVR or 0
;           No other registers are used

```

```

dox_reg    reg        d1/a2/a4
do_execb   movem.l    dox_reg, -(sp)
           movea.w    ri_exec, a2
           moveq      #0, d0

dox5       move.b     (a3)+, d0
           beq        dox1          end
           cmpi.b     #$32, d0
           bls        dox2          not p_.. or g_..

           moveq      #0, d1
           move.b     (a3)+, d1     w_.. to D1.W
           bra        dox3

dox4       adda.w     #204, a4      to next data space
dox3       dbf        d1, dox4
dox2       jsr        (a2)         do operation in D0
           bne        dox1          ---->      OVR
           movea.l    8(sp), a4     replace A4
           bra        dox5          -> next operation

dox1       movem.l    (sp)+, dox_reg
           rts

```



## ***Final Comment***

The extended system allows values of  $w_{...}$  to be from 0 to 255, giving at most 256 data stacks.

The maximum will accommodate  $256 \times 34 = 8704$  floating point numbers. This, in turn, allows for the storing of three matrices of size  $53 \times 53$ .

This may not satisfy those who wish to play with even larger matrices. What of size  $1000 \times 1000$ ? What indeed!

## **SANDY ELECTRONICS UPDATE PT 2      DAVE PARK**

(In the previous issue Dave was discussing problems and solutions to the QL expansion components - Ed.)

**W**e bounced back and forth, discussing which items I already had in stock, and a plan was devised. We could implement 896K of fast static RAM, shadowing the QL's internal RAM and screen RAM for maximum speed-up, for just a 15-20% increase in cost. What's more, we could add a through connector, and trick any older simple interface (we're not talking Gold Cards here!) by mapping its ROM into a location we chose.

By including a 4-position DIP switch, the owner would be able to configure 512K, 640K or 896K. They would also be able to select how the ROM of any attached interface would be treated. What's more, the RAM would operate at the same speed as ROM, so TK2 would work properly in timing-critical network situations.

The exciting thing about this is it allows you to turn almost any

floppy interface into a “Super Trump Card”. It doesn’t just have to be floppy interfaces either.

This is a very simple expansion that really benefits from having Minerva 1.98 or greater. Development is almost complete. We expect the “Sandy SuperMem” to be available around the time the next QUANTA magazine comes out.

UltimIDE has seen some progress. There’s now a full schematic, all the custom logic is done, and the circuit board’s development is well under way. From a hardware point of view, this will support a lot of devices, including some that haven’t been paired with a QL before. The current spec is for a master and slave PATA hard drive externally, two compact flash slots internally, plus an internal location for a 1.8” SSD. The internal format of the drivers is 100% compatible with the current QubIDE format, which should please a few people.

UltimIDE will be supplied with either a 512 MB or 1 GB CompactFlash card fitted internally; pre formatted to contain assorted utilities, drivers, the flash utility and some public domain content you can enjoy or delete at your leisure.

From a firmware point of view, the new 3.00 firmware has been developed from the 2.XX drivers, which have been released under the GPL open source license. When released, we will make the full driver and sources available. We will also make the new driver available to owners of existing QubIDEs, and we will make V2 GALs available to owners who have V1 GALs - all for just a nominal fee.

I have spent the last few days deeply exploring the new power

supply planned for UltimIDE. It will also be used on the Q-Regulator board - a 7805 voltage regulator replacement for the QL that will reduce heat and power draw, yet also leave a lot more power available for the QL and expansions.

I am testing the power supply well beyond its capabilities to ensure it truly is rock solid and unconditionally stable, since it will be the standard power configuration on our products going forward. It is very much cooler than the current 7805, and removes the hot-spot behind the Microdrives. It also provides much cleaner power and reduces the load on the external power supply by between 35-40%, with 2 amps of 5 volt power available to the QL.

The project that follows behind UltimIDE is UltraQ. UltraQ incorporates all the features of UltimIDE, but adds the following: floppy support via a WD1772, parallel, QIMI mouse, and memory.

We're still considering options to incorporate Ethernet and have a target device in mind. However, the general lack of response to our request on the ql-users mailing list has left us discouraged. We did get a few expressions of interest, but nobody came forward with a specific proposal or direction. We're still open to any approaches of assistance, and a developer would get beta access and finished hardware they'd get to keep.

The one killer feature of UltraQ will be memory. The QL uses a 68008 with 20 address lines, giving a 1 megabyte address range. We have sourced a batch of Freescale-made 68008FN CPUs which are 100% compatible with the 68008, but which have two extra address lines, giving an address range of 4 megabytes. By replacing the QL's CPU with our own, Gold Card style, we can give the base QL 4MB of RAM, the same as a Super Gold Card.

Once we make the commitment to replace the CPU, it makes sense to use some custom logic to give the QL some SGC-like features, including video RAM shadowing. We can also use some custom logic to increase the CPU speed quite considerably over a stock QL without causing problems for video - these Freescale CPUs clock at 15MHz quite happily.

Instead of using EPROMs, the UltraQ will store ROM driver images for IDE, floppy, etc. in flash memory, which will be mapped into the top of the 4 megabytes of memory. These drivers in flash will be in-system updatable using a supplied flash utility. One internal debate we're having is the possibility of including the operating system in flash.

Minerva 2.00 would be revised to accommodate the new memory map, multiple mirrored video screens, etc. There would be a number of spare 16K flash slots, and the flash utility would let you put TK2 or ICE ROM or etc. in those locations or SMSQ/E.

While we see UltimIDE arriving in Q3/14, we don't see UltraQ arriving until much later in the year. I think the QL will still be 30, hopefully, but the complexity of UltraQ will mean a lot of testing, prototyping, response and revisions.

It is important for me to moderate expectations on delivery: UltraQ's development is being funded from UltimIDE sales. We don't know how big the market is, so we'll be using UltimIDE sales to gauge the likely demand for UltraQ.

That's a detailed overview of current work here at Sandy Electronics. If there's interest, we are happy to report regularly on our progress.

QUANTA Membership Secretary's Updated Report  
for Year 2013

As at 27<sup>th</sup> April 2014 the QUANTA Membership was made up as follows: (2013 year-end figures are given in brackets for comparison). Figures given as nnE denote the number of members taking the Electronic version of the Magazine. All other members take the Printed copy which is posted to their home address.

	UK		OS	
Full Members	65	28E (75 36E)	20	15E (22 17E)
Associate Members	6	6E (6 6E)	Nil	(Nil)
Honorary Members	6	2E (6 2E)	2	(2)
Committee Members	8	3E (8 3E)	1	1E (Nil)
<b>TOTALS</b>	<b>85</b>	<b>39E (95 47E)</b>	<b>23</b>	<b>16E (24 17E)</b>
<b>Combined Total</b>	<b>108</b>	<b>55E (119 64E)</b>		

Records show that we have been notified that 2 of our members have resigned.

9 members have paid only part of their subscription and are additional to the above figures. None of these members have responded in any way to letters sent out to them. Their membership status is marked as "Lapsed".

We have had 13 new members since January 2013 (16 Months)

John Gilpin  
Honorary Membership Secretary.

At a QUANTA Committee Meeting immediately following the AGM, the cost per copy of the printed version of the QUANTA Magazine was discussed and the question asked:-

At what point do we consider the cost per printed magazine to be uneconomic and what solution should we take?

To consider the costs of purchasing our own colour laser printer, stapler and guillotine, and printing our own magazine.

OR

To consider making the QUANTA Magazine an eMag only – How many members do not have ANY facilities to download and read a .pdf file? (please send your views to:- Lee Privett – Magazine Editor - > [editor@quanta.org.uk](mailto:editor@quanta.org.uk) or by ordinary mail, the address is on page 2, Ed. < )

This will help us to make these decisions on behalf of the members.

**QUANTA HELPLINE**

**DILWYN JONES**

---

**M**embers wishing to submit helpline requests via email can use the email address [helpline@quanta.org.uk](mailto:helpline@quanta.org.uk) or if you prefer to use traditional post, please send the helpline request to me via the address printed inside the front cover of the magazine.

Obviously, we cannot guarantee to answer every query we receive, but we will do our best! Where we have been unable to answer the

queries, we may print the help request as an open request in the magazine to ask if any of the readers can come up with a solution. Of course, if readers feel that they have a better solution than we came up with, or would like to correct any errors we make, please write to us!

**Q I have started using QPC2. This version, as in previous versions, does not accept AUTO 100,10. The message is INVALID PARAMETER. When the QL first came out I used AUTO frequently. I use it as the first instruction when writing a program.**

A. In fact, strictly speaking, this is not down to QPC as such, more of a change in the way the SMSQ/E operating system (or even more specifically, the SBASIC interpreter) handles program entry and editing.

In SMSQ/E, or more correctly SBASIC, the AUTO command does not exist in the way in which it originally did in the original QL SuperBASIC.

SBASIC accepts the keyword AUTO, but implements it like the ED command in Toolkit 2 – it starts editing at the specified line number. So you can only give one parameter (the line number), like you can with ED. Here is a quote from the SuperBASIC Reference Guide, which explains this:

***AUTO - SMS NOTE:***

*On current versions of SMS, AUTO has been re-coded to be the same as ED, therefore it will not allow a second parameter, and merely places you in ED mode with the cursor at the specified start line number .*

While this may be awkward at first, I suppose the idea is that the ED command gives you a lot more facilities and flexibility.

**Q. Is it possible to change the system fonts, so that I can give all my programs a different look? I am aware of the CHAR\_USE command to set a different font in a program, but it would be nice to have all programs using the same font.**

A. Sadly, this isn't possible in original QDOS as the main system font is held in the ROM and cannot be changed. The designers of SMSQ/E, however, foresaw this requirement and added a command called CHAR\_DEF to the SBASIC interpreter.

Each channel can have two fonts, one of which normally covers

## RWAP Software

**Replacement QL Keyboard Membranes**

**Special Offer for Quanta Members  
£12 each plus post and packing**

**Quote "Quanta 2012 Offer"**


Post and packing:  
UK £2.50  
Europe £2.80  
Rest of World £3

Visit the QL Wiki for more details on software, books and hardware for the Sinclair QL:  
[www.rwapadventures.com/ql\\_wiki](http://www.rwapadventures.com/ql_wiki)

**RWAP Software**  
3 Dale View Court, Fulford, Stoke-on-Trent  
Staffordshire ST11 9BA  
Tel: 01782 398143  
Email: [rich@rwapservices.co.uk](mailto:rich@rwapservices.co.uk)

UK Cheques payable to 'R. Mellor' or use PayPal.

Website:  
[www.rwapsoftware.co.uk](http://www.rwapsoftware.co.uk) or [www.sellmyretro.com](http://www.sellmyretro.com)



Why a bad keyboard membrane could change your whole career strategy!

Working to safeguard the QL for the future. [www.rwapsoftware.co.uk](http://www.rwapsoftware.co.uk)



the usual ASCII range of characters up to code 127 and the other which covers the extended character set including accented characters and symbols with higher ASCII codes.

You can use any standard QL system font, there are plenty out there to download free on the internet from sources such as the Fonts page on my website at:

<http://www.dilwyn.me.uk/fonts/index.html>

You can also create your own, using one of the many font editors out there.

Before anyone says anything to correct me, Sinclair and most QL software developers have traditionally used the term “fount” instead of “font”.

To change the ‘system fonts’ used in SMSQ/E, we use a command called CHAR\_DEF:

```
CHAR_DEF font1_address, font2_address
```

As you might expect, the two addresses simply point to the address of a font you have loaded into the common heap memory. To do this, you would test the length of the font file with the FLEN function, use the ALCHP command to allocate that amount of memory (best to add the two lengths together into one block and load the fonts one after the other), then use CHAR\_DEF to tell the system where these fonts are.

There are two special values which can be used in place of the address values:

0 = go back to using the ones built into the operating system

-1 = don't change this setting, useful if you only wanted to change one of the two fonts, for example.

So, CHAR\_DEF 0,0 resets both character sets to that built into the operating system. CHAR\_DEF 0,-1 would reset the first (the one including letters, numbers etc) to that built into the system, while still using the existing second font you loaded.

Here is how to load two fonts.

```
100 REMark set filenames of the two fonts to be
used
110 font1$ = 'win1_fonted_serif_font'
120 font2$ = 'win1_fonted_font2_fnt'
130 fontlength1% = FLEN(\font1$)
140 REMark ensure it's an even value
150 IF (fontlength1% MOD 2)=1 THEN fontlength1% =
fontlength1%+1
160 fontlength2% = FLEN(\font2$)
170 :
180 REMark allocate memory to hold the fonts
190 fontladdress = ALCHP(fontlength1%
+fontlength2%)
200 IF fontladdress <= 0 THEN REPORT
fontladdress : STOP
210 font2address = fontladdress+fontlength1%
220 :
```

```
230 REMark load fonts
240 LBYTES font1$,font1address
250 LBYTES font2$,font2address
260 :
270 REMark set the new system fonts
280 CHAR_DEF font1address,font2address
290 :
```

By now, you may have tried running this program and realised nothing has changed if you are using the SBASIC windows #0, #1 and #2. This is because channels already open don't automatically use the new fonts for various reasons, the most obvious being that if the font loading failed, or you accidentally loaded a file of rubbish data, you would no longer be able to see what you are typing in as the letters, numbers and symbols might turn to garbage.

So, the safest thing to do is to open a new channel, such as OPEN #3,scr and test the printed output there with something like LIST #3. If it works as we expect, CLOSE #3 and next we use the original CHAR\_USE command to set the fonts for the existing channels:

```
FOR chan = 0 TO 2 : CHAR_USE
#chan,font1address,font2address
```

Indeed, this can hold true for programs which have already been started before you load the new fonts. The answer is simple – install the fonts first, before you do anything else such as load other programs, e.g. install the new fonts in your boot programs.

Used in this way, even “front end” programs like Launchpad can take on a new appearance using these new system fonts, as long as the fonts are installed before such programs are started.

You may have come across fonts which include all characters of the first and second QL fonts, some of the ones on my website for example – usually these fonts cover the full range of character codes from 31 (the default chequerboard characters) right up to 191, one of the arrow symbols.

To use these simply requires a minor bit of lateral thinking – we load the font as font1 and simply set font 2 to 0. This means that all valid characters will now be printed using the new font 1, but any out of range character codes can still be displayed using the default chequerboard character:

```
CHAR_DEF font1address,0
```

When resetting the fonts back to normal with CHAR\_DEF 0,0 you should remember to deallocate any heap area used for the new fonts, e.g. with the command RECHP font1address if you used the program above:

```
CHAR_DEF 0,0 : RECHP font1address
```

It is best to do this with no programs running, as the window channels may not spot the change and carry on trying to use the new system fonts which are no longer there.

Channels such as the SBASIC windows may not realise you have changed the default system font back to normal either. You have to use CHAR\_USE commands again to reset them back to normal,

but make sure you have used CHAR\_DEF first, otherwise CHAR\_USE might end up using the new fonts, since CHAR\_DEF is still pointing to them! Confused? I certainly was when I first issued these commands in the wrong order!!!

```
FOR chan = 0 TO 2 : CHAR_USE #chan,0,0
```

**Q. Is it possible to change the cursor shown by the QL? I'm fed up of just a flashing red rectangle?**

A. Not on a QDOS QL, but recent versions of SMSQ/E can do this, by allowing a sprite to be loaded and used as a cursor. Recent versions of SBASIC have new keywords which allow you to load a sprite of 6 pixels across and 10 pixels down.

CURSPRLOAD 'filename' is all that all is required once you have a suitable sprite. You can create 6x10 pixel sprites with most QL sprite editors such as Easysprite. For example, run this program to build a little green diamond shape sprite to use in place of the standard cursor.

```
100 f$ = "ram1_diamond_spr"  
110 fl = 104  
120 base = ALCHP(104)  
130 RESTORE  
140 FOR a = 0 TO fl-1  
150   READ byte  
160   POKE base+a,byte  
170 END FOR a  
180 SBYTES ram1_diamond_spr,base,fl  
190 RECHP base  
200 PRINT f$;' created.'
```

```
210 STOP
220 :
230 DATA 1,0,0,0,0,6,0,10,0,0
240 DATA 0,0,0,0,0,12,0,0,0,48
250 DATA 0,0,0,0,48,0,0,0,48,0
260 DATA 0,0,120,0,0,0,120,0,0,0
270 DATA 252,0,0,0,252,0,0,0,120,0
280 DATA 0,0,120,0,0,0,48,0,0,0
290 DATA 48,0,0,0,48,48,0,0,48,48
300 DATA 0,0,120,120,0,0,120,120,0,0
310 DATA 252,252,0,0,252,252,0,0,120,120
320 DATA 0,0,120,120,0,0,48,48,0,0
330 DATA 1,0,0,0
```

Once you've run this little program to generate a 104 byte sprite called `diamond_spr` in ramdisc `ram1_`, enter the command `CURSPRLOAD 'ram1_diamond_spr'` to set the sprite as SBASIC's new cursor.

There are two other handy extensions for turning the sprite cursor off or on, and not just for BASIC. By giving either the job name, or job number and job tag (you can see these by listing the running job with the `JOBS` command).

```
CURSPRON "jobname"
```

or

```
CURSPRON jobnum,jobtag
```

turn on the sprite cursor, while

```
CURSPROFF "jobname"
```

or

CURSPROFF jobnum,jobtag

turn off the sprite cursor. To allow a job such as SBASIC to turn off its own cursor sprite, just issue the command CURSPROFF -1.

Using these commands you can replace the flashing cursor for many programs as long as you know their details, for example, to replace the Xchange cursor: CURSPRON "Xchange"

The sprite may include transparent pixels, that's how I was able to cut out the corners to create the diamond shape, so you could (theoretically at least) create a rounded cursor, or a hollow box, a cross, a line or even just a dot, probably even an invisible 6x10 cursor although I didn't dare try that one!

If you change the cursor for a pointer driven program, it doesn't replace the standard pointer arrow, just the flashing cursor you get when asked to INPUT something, for example. So if you were to try to change the cursor for Launchpad or such a program, it would continue to use its standard pointer arrow for most things, only using the new cursor sprite when asking you to enter something from the keyboard.

**Q. When I used a QL, it was easy enough to save a copy of the screen with the command SBYTES filename,131072,32768. On modern systems like QPC2 the screen can be bigger and more colourful. How can I work out how to save one of these bigger screens – I'm told they can be longer and stored in a different place in memory.**

A. As you have realised, saving a screen picture is a bit more

complex on SMSQ/E systems since the screen size can vary, and its address in memory can change too. Sorry everyone, this will be a bit of a long answer!

The good news is that SBASIC has extensions which can help greatly to simplify saving screen pictures.

In order to save a screen, we need to know

- Its base address in memory
- How many pixels across
- How many pixels down
- The length of each line in bytes

To find where the screen starts in memory, we use the SCR\_BASE function:

```
LET base_address = SCR_BASE
```

To find how many bytes between the start of one line and the start of the next, SBASIC provides the SCR\_LLEN function (standing for Screen Line Length).

```
LET line_length = SCR_LLEN
```

The two functions SCR\_XLIM and SCR\_YLIM tell us the screen x limit and screen y limits. Used without a channel number, or with the channel number of the lowest open channel (the primary channel), they tell us how many pixels across and down the screen respectively.

```
LET across = SCR_XLIM
```



```
LET down = SCR_YLIM
```

So, adding the information from these functions to an SBYTES command, we can now save a screen in any mode like this, using this simple two line program from the SMSQ/E manual:

```
ssz = SCR_LLEN * SCR_YLIM : REM screen size
SBYTES filename$, SCR_BASE, ssz : REM save
the screen
```

You could easily combine them into one line, like this:

```
SBYTES filename$, SCR_BASE,
SCR_LLEN*SCR_YLIM
```

And to reload the screen later needs just one simple command, as long as you know that the file is the right size for the screen mode and resolution you are using:

```
LBYTES filename$, SCR_BASE
```

As ever, though, there is a minor fly in the ointment. The Aurora card. Some of its display modes surprisingly do not use the full line length! The Aurora is always geared up to have a line length wide enough to hold 1024 pixel wide screens in 4 colour mode, a fixed line length of 256 bytes per line. Saving and reloading will probably work OK as long as you are loading the screen back on the same computer.

If you used the above program line to save the screen, it would seem to work, but it would add the unused memory space to the right of the screen, so you might get a bigger screen file than you

expected and it might either have a blank or random section to the right of the picture.

If you wish to save a screen comprising of just the actual visible line length, things become a little more complex, but not impossible by any means. What you have to do is to calculate how wide each line should be, and save that amount from each Aurora video line in memory, line by line, using the PEEK\$ function to fetch the relevant part of the video of each line.

To calculate the actual length of the visible line, we have to know how many bits or bytes correspond to each pixel and save the right number of bytes for each line.

In MODE 4 and MODE 8, we divide the number of pixels across by 4 to get the number of bytes. In theory we have to make sure this rounds up to an even number of bytes. In practice, it always does for a full screen.

MODE 4 and MODE 8: 8 pixels per two bytes.

```
LET visible_length = SCR_XLIM DIV 4
```

256 colour (8-bit) screen mode 16: One byte per pixel

```
LET visible_length = SCR_XLIM
```

*To Be Continued...*

(Due to the length of this Helpline response, further information has been carried over to the next issue—Ed.)

**COPY DATE for AUG/SEP 2014 ISSUE is:  
5th AUG 2014**

**Please ensure that your copy reaches the editor on or before that date. Copy on Paper, Magnetic Media or by Email. Text & listings are preferred in ASCII format as text files, Microsoft™ Word Documents or in QL Quill formatted files. All graphic submissions in as higher resolution as possible, preferably in PNG format, other forms of submission can also be accepted, please contact the Editor for details.**

**All Contributions are Welcome  
Copy by Email to - [editor@quanta.org.uk](mailto:editor@quanta.org.uk)**

*This magazine and all the articles etc. within it are copyright © QUANTA 2014 unless otherwise stated. Reproduction and redistribution in any form is not allowed without the express written permission of the publisher. All copyrights and trademarks are hereby acknowledged.*

# The online QL community

Now with QUANTA section!

New online chat facility!

Free to join!

No registration  
needed to view

Please stop by for a visit!



[www.qlforum.co.uk](http://www.qlforum.co.uk)